AMENDMENTS TO THE CLAIMS

Claims Pending:

- At time of the Action: Claims 1-29
- Amended Claims: Claims 1 and 14
- After this Response: Claims 1-29

1. (Currently Amended) A method implemented at least in part by a machine, the method comprising:

receiving an input that includes hierarchical data;

traversing a filter tree according to segments of the hierarchical data to locate one or more matching nodes that correspond to the hierarchical data;

comparing at least a portion of the input to one or more filters associated with the matching nodes; and

executing instructions associated with one or more filters satisfied by the input.

2. (Original) The method as recited in claim 1, further comprising applying a precedence rule to the matching nodes, wherein the comparing step is performed on a subset of the matching nodes that is determined by the precedence rule.

3. (Original) The method as recited in claim 1, wherein a node may be associated with instructions but no filter, in which case the input is deemed to match a filter for the node, thereby resulting in execution of the instructions.

4. (Original) The method as recited in claim 1, wherein the hierarchical data

further comprises a path that identifies a location in a hierarchical system.

5. (Original) The method as recited in claim 1, wherein:

a root node of the tree corresponds to a first segment of the hierarchical data;

a child node of the root node corresponds to a second segment of the hierarchical data; and

a bottom-level node of the tree corresponds to a last segment of the hierarchical data.

6. (Original) The method as recited in claim 1, wherein each node of the filter tree references zero or more filters.

7. (Original) The method as recited in claim 1, wherein the hierarchical data further comprises a destination path identified by a segment string.

8. (Original) The method as recited in claim 7, wherein the input further comprises message data that is transmitted to a location identified by the destination path if the input satisfies a filter that is referenced by a filter tree node associated with the destination path.

9. (Original) The method as recited in claim 1, further comprising:

identifying the hierarchical data contained in the input; and

parsing the hierarchical data into segments for use with matching.

10. (Original) The method as recited in claim 1, wherein the traversing step further comprises:

comparing a first segment of the hierarchical data with a first node in a filter tree level that corresponds with a position of the first segment in the hierarchical data;

if the first segment does not match the first node, determining that the input does not match the first node;

if the first segment matches the first node and there is a subsequent second segment in the input, comparing the subsequent second segment to one or more second nodes in the filter tree that are subordinate to the first node; and

if the first segment matches the first node and there is not a subsequent second segment in the input, determining that the input matches the first node.

11. (Original) A filter tree data structure stored on one or more computer-readable media, comprising:

a first level having a root node that corresponds to an initial segment of hierarchical data;

at least one intermediate level having at least an intermediate node that corresponds to an intermediate segment of the hierarchical data, the intermediate node being subordinate to the root node;

a bottom level having at least a bottom level node that corresponds to a final segment of the hierarchical data, the bottom level node being subordinate to an intermediate node; and

wherein at least one node is an active node that references an instruction set that is executed when an input is received that includes the segment corresponding to the active node and segments corresponding to all nodes superior to the active node.

12. (Original) The filter tree as recited in claim 11, wherein:

the active node further comprises a reference to a filter; and

the instruction set is executed only if the input satisfies the filter.

13. (Original) The filter tree as recited in claim 11, wherein the instruction set if executed only if the active node is a bottom level node.

14. (Currently Amended) A system, comprising:

memory;

a filter tree stored in the memory, at least one node of the filter tree referencing a filter;

a message input module configured to receive a message of hierarchical data that includes a path having one or more segments of hierarchical data;

a primary matching module configured to locate one or more filter tree nodes that match one or more of the segments;

a secondary matching module configured to identify any filters associated with the one or more matching filter tree nodes and to compare the message against the filters to determine if the message satisfies any of the filters; and

a message processing module configured to execute instructions associated with any filter that is satisfied by the message.

15. (Original) The system as recited in claim 14, wherein the secondary matching module is further configured to apply a precedence rule to the one or more matching filter tree nodes to derive a subset of the matching filter tree nodes and to identify only the filters associated with the subset of matching filter tree nodes.

16. (Original) The system as recited in claim 14, wherein the filter referenced by the at least one node further comprises a null filter that is deemed to be satisfied by any input message compared thereto, thereby resulting in the execution of instructions associated with the node referencing the null filter.

17. (Original) The system as recited in claim 14, wherein the message path is of a hierarchical nature and successive path segments correspond to successively subordinate levels of the filter tree.

18. (Original) The system as recited in claim 14, wherein the primary matching module is further configured to:

identify one or more path segments included in the message;

locate filter tree nodes associated with each path segment; and

determine that each node located matches the message.

19. (Original) The system as recited in claim 14, wherein the primary matching module is further configured to:

identify one or more path segments included in the message;

locate filter tree nodes associated with each path segment; and

determine that a node associated with a final path segment is the only node that matches the message.

20. (Original) The system as recited in claim 14, wherein the primary matching module is further configured to:

identify one or more path segments included in the message;

locate filter tree nodes associated with each path segment; and

determine that a node associated with an initial path segment is the only node that matches the message.

21. (Original) A method, comprising:

receiving a data transmission requesting to add an entry to memory that stores multiple filters, the data transmission including at least a new filter, a segmented path identifying hierarchical reference data associated with the new filter, and at least one data item associated with the new filter;

traversing a hierarchical data structure stored in the memory that is used to reference each of the multiple filters to determine if an existing location in the data structure matches the segment path included in the data transmission;

if an existing location is identified, adding a reference to the new filter to the existing location; and

if an existing location is not found, creating a new location in the data structure, the new location being determined according to the hierarchical reference data and storing a reference to the new filter at the new location.

22. (Original) The method as recited in claim 21, further comprising storing a reference to the data item at the new location so that an input matching the new location can access the data item.

23. (Original) The method as recited in claim 21, wherein:

each of the multiple filters stored in the memory is associated with hierarchical reference data; and

the memory includes a filter tree data structure to reference the multiple filters according to the hierarchical reference data, each of the multiple filters being associated with a node in the filter tree data structure.

24. (Original) The method as recited in claim 23, wherein:

each segment of the segmented path corresponds to a node in the filter tree data structure so that a node associated with a filter corresponds with a final segment of the segmented path; and

each of one or more superior nodes to the node associated with the filter corresponds with a preceding segment of the segmented path.

25. (Original) One or more computer-readable media containing computer-executable instructions that, when executed on a computer, perform the following steps:

receiving a data transmission that includes at least a message and a destination path that is structured according to a hierarchical protocol;

parsing the destination path into constituent segments;

comparing the destination path segments to a hierarchical data structure that is arranged according to the hierarchical protocol to determine one or more matching locations in the data structure that correspond with the destination path;

identifying one or more filters associated with the one or more matching locations;

testing the message against one or more filters referenced by the one or more matching locations; and

for each filter satisfied by the message, executing one or more instructions associated with the filter.

26. (Original) The one or more computer-readable media as recited in claim 25, wherein the hierarchical protocol is eXtensible Markup Language (XML) protocol.

27. (Original) The one or more computer-readable media as recited in claim 25, wherein the one or more filters are defined using Xpath.

28. (Original) The one or more computer-readable media as recited in claim 25, wherein the one or more instructions associated with the filter further comprise a

destination to which at least a portion of the message is transmitted.

29. (Original) The one or more computer-readable media as recited in claim 25, wherein the destination path is a network address.